

CUIで始める動画変換

61 回生 中田亮

前書き

どうやら今年の部誌は量・質ともに不足気味らしくて、それを補う為に何故か私が部誌を書くという事態になっています。実は私は本来部員ではないのですが、部室のサーバや壊れたマシンを復旧したりしているうちに部員の一人として認識されてしまい、今年の部誌の量・質を補う為に何か書いてね〜と相成った訳で...。数年前はパソコン部員の技術もそれなりに高かったはずですが、技術が伝承されておらず、どうも質の良い物が書けていないようです。この部誌も昔の先輩方を引っ張りだして作っているようですね。この先パソコン部大丈夫なんでしょうか。しかし、そんなことを嘆いてもしょうがない。今年は「動画変換」について書く人がいないそうなので、私が書くことにしました。ちなみに、当初は初級編・中級編・上級編と3つに分けて、最初はGUIから入って少しずつCUIに慣れられていく、という方式をとっていたのですが、逐一解説していくと流石に長くなりすぎるので上級編だけにしました(苦笑) そのためかなりマニアックな内容になっていることは否めません。それに、生半可な知識で書いているので見る人が見ればかなり恥ずかしい内容になっているかと思いますが...まあ部誌の量が稼げるわけですからそれでもいいでしょう(苦笑)

エンコードの前に

さて、前年度の部誌で動画変換について書かれている物と言えば、「録画番組パソコンで！AVI版。」計画再浮上！、ですね。去年の部誌をなくした人はnpcaのホームページにあるのでそちらを読んでください。

ここでは動画の変換の基本方針として、「タダ」「軽い」「綺麗」「速い」「人気」の5つをあげています。しかし今回は少し違います。今回の変換の基本方針は画質です。いかにして少ないビットレートで高画質にするか。それが動画変換の永遠のテーマなわけで、今回はその限界に挑んでいこうと思います。去年の部誌では、適当なエンコードとして「divx」「xvid」「wmv」「x264」の4つが紹介されていましたが、画質を取るならx264で決まりです。ちなみに、snowという、x264よりもさらに先進的なエンコードもあるようですが、再生互換性や再生負荷のことを考えるとちょっと....。互換性はともかく、再生負荷重すぎです。H.264もさんざん重いと言われましたが、それよりも更に重いです。まあ、使えない訳ではないので、さらっと説明しておきます。

また、画質を語る上で重要なのはエンコードの性能だけではありません。ビデオフィルタも重要です。変換前にノイズリダクション、インターレースの解除等を行う必要があります。この性能にかかわってくるのがビデオフィルタです。ノイズリダクションやインターレースの解除をうまくできるか否かで、画質がかなり違ってきます。これも

重要なポイントです。ビデオフィルタの機能を持ち合わせているソフトは、「avisynth」「mencoder」「aviutl」「ffmpeg」等があるのですが…。ここでちょっと言うことがあります。私は Macintosh 使いなので、Macintosh を前提として話を進めていきます。ああ、この時点で更にマニアックになった気がしますが…。気にしないで進めましょう。まあ、そんなにプラットフォームに依存することは書いてありませんから、Windows や Linux でも多少手順が違うだけで応用が効くかと思います。また、Mac OS 9 以前をお使いの方は残念ながらここに書いてあることは全く役に立ちません。Mac OS X にアップグレードすることをお勧めします。さて、aviutl に関しては、AVI 出力専用です。(プラグインを組み込めば AVI 以外にも出力できますが、実質 AVI 出力専用です) 以前は手軽に高画質が実現できるということで広く使われていたのですが、x264 が AVI(VFW) コンテナのサポートを廃止したので実質使えません。x264 開発ページに「no more vfw」の告知が出たときは結構騒がれたのですが、まあいずれにせよ aviutl は Windows でしか使えないので対岸の火事といったところですね。avisynth は、aviutl ほど手軽に使えるわけではありませんが、優秀なビデオフィルタです。Windows を使っているのであればこれを使うべきなのでしょうが、今回は Macintosh 向けに書いているので解説しません。さて、Macintosh で一番使えるビデオフィルタは、mencoder です。もともと UNIX のソフトウェアなので CUI から利用することになるわけです。ffmpeg は GUI で利用できる動画変換ソフトで内部的に使われているようですが、ビデオフィルタは貧弱です。ということで、Macintosh でビデオフィルタを選ぶなら mencoder に決まりです。

エンコード方法の解説

さて、早速 mencoder をダウンロードしてきましょう。本来 UNIX のソフトウェアはソースコードだけ置いてあって、勝手に各自ビルドして行ってください、といったノリのものが多いのですが、Mac OS X で動く mencoder のバイナリは有志がビルドしてくれたものがありますから始めのうちはそれを使いましょう。

<http://ffmpegx.com/download.html>

上記の URL は ffmpegX という GUI で動く動画変換のソフトですが、mencoder のバイナリが置いてありますのでそれを使いましょう。Windows、Linux のバイナリも探せばあるとは思いますがここでは紹介しません。

当然ですが CUI で動くツールなのでダブルクリックしても開きません。ターミナルから利用します。ここではターミナルの使い方については解説しません。ここに書くとな長くなりすぎるので。必要なら自分で調べてください。ところで、mencoder だけ落としてきて、肝心の x264 は要らないのかと思われるかもしれませんが、落としてきた mencoder の中には x264 も含まれているので問題ありません。

さて、早速変換してみましょう。ホームフォルダ直下に mencoder と変換したいファイル(ここでは hoge.mpeg としています)を置いて、アプリケーションフォルダ内にあるターミナルを開いて以下のコマンドを実行すれば変換できます。ターミナルの扱い方に付いて説明しだすと長くなるのでここでは割愛します。別に Macintosh でなくとも同じコマンドで変換できるはずです。

```
./mencoder hoge.mpeg -nosound -ovc x264 -x264encopts bitrate=1024  
:bframes=3 :b_adapt :weight_b :b_pyramid :keyint=240 :keyint_min=1
```

```

:scenecut=65 :qp_min=10 :qp_max=51 :qp_step=8 :qcomp=0.6 :ratetol=4
:deblock :deblock=0,0 :cqm=jvt :cabac :direct_pred=auto :nofast_pskip
:nodct_decimate :nointerlaced :noglobal_header :psnr :ssim :pass=1
:threads=2 :turbo=1 -passlogfile hoge.264.log
-vf pullup,softskip,pp=15,crop=720:480:0:0,scale=640:480:::3,
hqdn3d=4:3:6,harddup -sws 9 -ofps 24000/1001 -of rawvideo -o hoge.264

./mencoder hoge.mpeg -nosound -ovc x264 -x264encopts bitrate=1024
:bframes=3 :b_adapt :weight_b :b_pyramid :keyint=240 :keyint_min=1
:scenecut=65 :qp_min=10 :qp_max=51 :qp_step=8 :qcomp=0.6 :ratetol=4
:deblock :deblock=0,0 :cqm=jvt :cabac :direct_pred=auto :nofast_pskip
:nodct_decimate :nointerlaced :noglobal_header :psnr :ssim :pass=2
:threads=16 :me=umh :me_range=32 :subq=7 :frameref=4 :mixed_refs
:8x8dct :partitions=all :trellis=2 :brdo :bime -passlogfile hoge.264.log
-vf pullup,softskip,pp=15,crop=720:480:0:0,scale=640:480:::3,
hqdn3d=4:3:6,harddup -sws 9 -ofps 24000/1001 -of rawvideo -o hoge.264$

```

ageha さん¹⁾の設定です。感謝！（アニメ用設定ですので、実写では変えましょう。解説読んでください。）

な、長いですね…。流れるには、まず高速な設定で変換し、そこでできた映像ファイルは捨ててログを取得、それを使ってもう一度低速だが高画質な設定でエンコードします。いわゆる2パスエンコードですね。これから ageha さんの設定に基づいて、オプションを一つずつ解説していきます。

-nosound

サウンドを出力しない。映像を生データで出力し、音声を別途変換してMP4に詰めるのでここでは音声は変換しません。

-ovc x264

エンコーダの指定。x264 を使います。

-x264encopts

この後に続く長ったらしいオプションは x264 の設定項目です。x264 を mencoder を経由せず直接利用した場合は違った書式になるのですが、おおむね対応しているようです。

-passlogfile

ログファイルの場所を指定します。2パスエンコードで使います。

-vf

ビデオフィルタのオプションをこのあとに指定します。

-sws

スケーリング (ビデオのリサイズ) の設定。0~10 の間で指定します。数字が大きくなると、輪郭がなめらかにリサイズできますが、その分遅くなります。拡大する場合は結構差が出ますが、縮小・等倍の場合はあまり見た目変わりません。まあでも、そこまで遅くなるわけでもないのだから9になっているようです。

¹⁾<http://agehatype0.blog50.fc2.com/>

-ofps

出力 fps の指定。ビデオフィルタを經由し、インターレースを解除したので出力 24fps となります。正確には、23.976 ですが、より正確には $\frac{24000}{1001}$ です。ここでは分数で指定できるので $\frac{24000}{1001}$ になっています。

-of rawvideo

映像を生データで出力します。後で音声とともに MP4 に詰め直すため。

-o hoge.264

出力ファイル名の指定。

-x264encopts のオプションを解説します。ここを弄ることで画質が変わってきます。一部省略してます。

bitrate=1024

ビットレートを指定します。ここでは 1024 になっていますが、私は 768 でも十分だと思えます。ですが、画質にこだわる場合は 1024 が良いのでしょうか。

subq=7

1~7 の間で指定します。subpel 精製品質の調整、らしいですが、そんなこと言われても訳分かりません。単純に、品質の度合いを表す数字と受け取って構わないようです。数字をあげると品質は良くなりますが遅くなります。とはいうものの、この値が低いと効かないオプションもあるので、最低でも 5 以上にした方が良いでしょう。出来れば 7。

frameref=1 frameref=4

変換時に何枚前のフレームまでたどるか、という指標です。増やせば増やす程画質が良くなり速度が遅くなっていくのですが、5 あたりで効果が激減します。この値が多いと結構時間がかかるので、1 パス目は 1、2 パス目で 4、としているようです。アニメに効果的。

keyint=240 keyint_min=1

GOP の最大値、最小値を指定します。GOP とは Group Of Picture の略で、動画を圧縮する際には GOP 単位に区切って圧縮します。

GOP は、I フレーム、P フレーム、B フレームから構成されています。I フレームは、それだけでデコードが可能です。ようするに、jpeg や png の写真のように、それ単体で表示出来るようになっているフレームです。

P フレームは、デコードするのに直前の I フレームを必要とします。直前の I フレームとの「差分」だけを記録しているのです。その為、アニメの口パク等、ほとんど絵が変わらない場面では非常に圧縮効率が良いです。

B フレームは、更に P フレームの差分をとった物です。デコードするのに直前の I・P フレームを必要とします。使いすぎると画質が悪くなることもあります。よって、実際のデータの並び順としてはこんな感じになる訳です。IPPPBBIPPP-BIPBBBIPPPIP... (B フレームはデータの並び順とデコード順が違うのですが、ここでは説明を割愛します)

ここで、IPPPBB までが 1 つの GOP、また次の IPPPB がまた一つの GOP です。I フレームをシーンチェンジの時ににおいて、そのあと少しずつ画面が変わっていく場面で P/B フレーム、またシーンチェンジが訪れたら I フレームを挿入、としてやって効率良く圧縮していく訳です。

GOPの最大値に設定された値になると、シーンチェンジが訪れていなくても強制的にIフレームを挿入します。つまり、画質的にはマイナスです。しかし、最初からではなく任意の場所から再生したいときは、Iフレームからでないで頭出しが出来ませんからあまりGOPが長いと頭出しがしにくくなってしまいます。つまり、keyintの値は下げれば下げる程頭出しがしやすくなり、画質的にはマイナス、ということです。設定する値は240フレーム=10秒分で良いんじゃないかなと思います。頭出しなんか知ったことか！画質を追求する！という方はもの凄く大きくしてやれば望みうる最善画質になります。お勧めしませんが。

GOPの最小値は、この値よりも小さな値でシーンチェンジが訪れたとしても新しいGOPにしません。つまり、画質的にはマイナスです。(Pフレームを挿入する訳ではなく、IDRフレーム等が絡んでくるのですが、ややこしくなるので割愛します。)ですが、実写では一瞬だけ映像を表示して次の瞬間には全く違う映像を表示する、なんてことはあまりないため、最小値を設定しておいた方がより効率よくエンコードできます。keyint_min=30くらいでしょうか。まあ、バイオハザードアポカリプスの終盤とかフラッシュバック的に連続して映像が表示されることもあるので、そこんとこどうかな〜とはおもいますが。アニメでは一秒未満でシーンチェンジすることも珍しくないなので、keyint_min=0もしくは1です。

scenecut=65

シーンチェンジを判定してIフレームを挿入するのですが、どれくらい映像が変わったらIフレームを挿入するか、という指標です。アニメではシーンチェンジが多いので高めに。特にアニメのOPでは更に多くなりますのでこの値よりも更にもう少し高くしても良いでしょう。一方、実写では低めに。scenecut=40(デフォルト)くらい。落語など、ほとんど場面転換がない特殊な場合はさらに低くします。

bframe=3

連続するBフレームの最大数。例えばbframe=3だと、IPPBBBとなるのが最大でIPBBBBとはなりません。bframe=0にするとBフレームを使いません。画質を求めるならBフレームは使うべきです。

b_adapt

Bフレームの品質に影響するようです。とりあえず使つときましよう。当然bframe=0の場合は意味ありません。

weight_b

前述したようにBフレームは使いすぎると画質が悪くなります。このオプションをつけると、Bフレームに適さないフレームを発見しPフレームにします。bframe=1ならオフ、2以上ならオンで。

b_pyramid

BフレームはPフレームの差分をとったものですが、このオプションを使うと更にBフレームの差分をとります。その分圧縮効率はよくなります。bframe=2以上でないで意味ありません。

deblock

デブロックフィルタ。これは必ず使うべきです。x264のデブロックは大抵の場合、画質がよくなるのに対し速度はあまり低下しません。アニメに効果的。

deblock=0,0

デブロック強度の指定。-6~6の間で指定します。大抵の場合、0,0でベストな

画質になりますので、素材に忠実にしたい、と思っけていても下げるのはお勧めしません。あげるのは、オリジナルの段階で既にブロックノイズがある場合です。deblock=3,3 程度でしょう。

cabac

若干遅くなりますが、ビットレートが10%~15%節約できるので切らない方がいいです。画質には影響しません。

nofast_pskip

P フレームにおける速い段階でのスキップ検出。通常は画質低下等のペナルティ無しで速度が向上するのですが、切ることで x264 特有の「閾階調でばたばた動くブロックノイズ」を抑えることが出来るので、ここでは切っているようです。実際に自分で試してみても決めることをお勧めします。

nodct_decimate

使うと若干ディテイルを除去するので、余ったビットレートを他に回せます。ここでは切られていますが、低ビットレートで変換する時やアニメでは使った方がいいかもしれません。ある程度ビットレートを見積もって素材に忠実にしたい場合はオフで。

pass=1 pass=2

パスの指定。1 が1パス、2 が2パス、という意味ではありません。1 が最初のパス、2 が最後のパス、3 がそれ以外のパスです。ですから、たとえば5パスでエンコードしたい場合は13332と変換していくわけです。とはいうものの、5パスもやるともの凄く時間がかかりますし、x264のマルチパスはあまり質がよくなるようなので3パス程度にするくらいなら2パスにしておいた方がいいです。短編動画ではマルチパスが効果的です。短編故に時間もあまりかからないことですし。

threads=2 threads=16

複数CPUで効率的に変換する為にスレッドを分割。最近、CoreDuoなど、マルチコアのCPUが登場してきましたが(もっともそれより前からマルチコアのCPUはあったのですが)マルチコアのCPUで動画変換を行うと、(2コアなら)2コアのうち1コアしか変換作業をせず、もう片方のCPUが宙ぶらりんな状態になります。それだと非常に勿体ないので、ここで指定された数値だけ作業を分担。複数のCPUに作業させます。分担させると圧縮効率に若干悪影響があるのでシングルコアのCPUを使っているならこのはthreads=1にしましょう。agehaさんのサイトでは、デュアルコアの場合で1パス目はthreads=2、2パス目はthreads=16とするのがもっとも良いという結論に達しています。最近はクアッドコア(4コア)やオクトコア(8コア)のワークステーション(もはやパソコンとは呼ばない)も登場してきました。きっとこれに動画変換をやらせたら気持ちが悪いくらい速くなるんでしょうねえ...

turbo=1

ターボモード。1パスで有効。あまり画質に影響しない1パスのオプションを切って、エンコード速度を速くします。0~2で指定します。1が良いでしょう。0よりは1の方が、2はちょっとオプション切りすぎな気がします。

nointerlaced

映像をインターレースされている物として扱います。mencoderでビデオフィルタでインターレースが解除されてから圧縮するので、当然オフ。

`noglobal_header`

ヘッダ情報を書き込む。PSP など一部のプレイヤーで必要になります。正直どちらでもいいです。

`psnr ssim`

それぞれ素材に対する忠実度を測る指標です。指定すると変換終了時に PSNR 値、SSIM 値を表示します。

`me=umh`

動き補償アルゴリズムの選択。次の中から指定します。

`dia`

ダイヤモンド。四角形サーチ。

`hex`

ヘキサゴン。六角形サーチ。

`umh`

マルチヘキサゴン。不等複数六角形サーチ。

`esa`

エクゾースティッド。徹底サーチ。`esa` は実用に堪えない程遅いですので使い物になりません。2 パス目で `umh` を使うのがいいでしょう。

`me_range=32`

`me=umh,me=esa` で動きを探索する半径を指定します。これもあまり大きくしすぎると遅くなります。重いと思うなら `me_range=24` くらいに下げてもいいかもしれません。

`mixed_refs`

複数のフレームを参照することで動き補償の精度が向上します。オンにしておきましょう。

`partitions=all`

動き補償をする際、フレーム全体が同一方向に動くことは少なくであちらこちらに散らばった動き方をします。そこで、マクロブロックという区画に仕切るのですが、その大きさを指定します。

<code>p16x16,b16x16,i16x16</code>	常に適用
<code>p16x8,p8x16,p8x8</code>	<code>p8x8</code>
<code>p8x4,p4x8,p4x4</code>	<code>p4x4</code>
<code>b16x8,b8x16,b8x8</code>	<code>b8x8</code>
<code>i8x8</code>	<code>i8x8</code>
<code>i4x4</code>	<code>i4x4</code>

たとえば、`p16x16,b16x16,i16x16,i8x8,i4x4` の 5 つを使いたい場合は `partirions=i8x8,i4x4` という風に指定します。`all` を指定すると全て使用。`none` を指定すると `p16x16,b16x16,i16x16` 以外は全て使いません。これも ageha さんのサイトで、何も考えず `all` を指定してよさそう、という結論になっています。

`8x8dct`

`i8x8` を使うならオンにしておきましょう。

`-vf` オプションを解説します。インターレースの解除やノイズの除去に影響します。`pp` 系は基本的にインターレースをほぼ除去できます。

pp=1b

リニアブレンド。これを使うと完全にインターレースを除去できますが、全体がややボケて、動く物が二重化します。前後のフレームを半分ずつ混ぜ合わせた、ゴーストのようなフレームができます。

pp=15

pp=1b の弱くした物。pp=1b と比べて副作用が少ないですが、場合によってはインターレースが残ることも。

pp=md

中央値補間。pp 系の中では単体で使う分でもっとも良い。デメリットはアニメなどの輪郭にジャギが出ること。

filmdint=fast=0/comb_thres=48

pp=md よりもクリアで、輪郭ジャギも出ません。デメリットは細かい部分が少しつぶれ、解除もれが出ること。アニメの口パクなどで良く残ります。filmdint 内でもオプションがあり、いろいろ挙動が変更できるのですが、ここでは解説しません。Macintosh では 2005 年夏頃から正常動作しなくなったようです。yadif=0 を使いましょう。

pullup,softskip

逆テレシネフィルタ。正確にはインタレ解除ではありません。インターレースがかかる前の状態に復元することを念頭においているので、アニメでは極めて美しく解除できます。欠点は、インターレースをかけた上から字幕ロゴ等が合成してあるとその部分が荒れること、始めからインターレースのかかった状態で撮影されているような番組では当然正しく解除できません。pullup 直後の softskip は必ず指定しましょう。

yadif=0

解除性能は filmdint とほぼ同等。ただ、解除もれを起こさない点で filmdint に勝ります。

mcdeint=0

新しく登場してきたインタレ解除フィルタです。インタレ解除フィルタのくせに動き補償まで備えています。yadif よりジャギが少なく、設定次第で万能ナイフにもなりそうですがまだ出てきたばかりで未実装の機能等があります。あと、猛烈に遅いです。

hqdn3d=4:3:6

デノイズフィルタ。前から順に、空間軸輝度、空間軸彩度、時間軸のデノイズ強度です。hqdn3d=4:3:6 がもっとも良い、という結論になっています。DVD など、ノイズが入っていない素材では削ってもいいでしょう。

crop=720:480:0:0

クロップ。映像の上下左右を裁ち落とします。上下の黒幕を裁ち落とす際に使います。最初のふたつで残す映像の幅と高さを、あとのふたつで切抜きの開始位置を指定します。注意したいのは、一般的な MPEG2 動画は 720 × 480 になっています。再生時に、640 × 480 に縮小しているのです。ですから、そこを留意した上で切りぬぎましょう。

scale=640:480:::3 スケーリング。上の crop で裁ち落とした映像を指定したサイズにリサイズします。これを省くと、リサイズを行わない、つまり crop してい

ない場合は 720 × 480 になって横に延びた映像になるので必ず指定しましょう。最後の 3 は、リサイズする際に輪郭の滑らかさなどに影響します。アニメなら 3、実写なら 4 が適切。

`harddup`

`x264` で変換する際は必ず指定しましょう。でないと音ズレします。必ず `vf` オプションの最後に指定すること。

変換が終了すると、拡張子が 264 のファイルが出来あがります。これはこのままでは再生できません。MP4 に映像と音声を入れる必要があります。それには MP4Box を使用します。MP4Box のバイナリは以下のサイトで取得できます。今度は Windows や Linux のバイナリも入っています。

<http://www.tkn.tu-berlin.de/research/evalvid/>

MP4Box をホームフォルダ直下において以下のコマンドを実行します。

```
./MP4Box -fps 23.976025 -add hoge.264 -add 音声ファイル.mp4 -new hoge.mp4
```

音声ファイルは、AAC のものをつかいます。MPEG2 ファイルから音声を抽出するには MPEG Streamclip²⁾を使います。抽出したあとは普通に iTunes 等で変換すれば良いです。動画ファイルから音声を抽出するには、拡張子は m4a になると思いますが、mp4 に変更しておきましょう。-fps は出力 fps です。小数で指定しないと音ズレします。

コマンドを実行し終わると、hoge.mp4 ファイルを生産します。これで完成です。お疲れさまでした。他のファイルは必要ないので削除しても構いません。なお、QuickTime では再生できない場合があります。VLC³⁾や MPlayer⁴⁾といった再生ソフトを使いましょう。ちなみに、mencoder は mplayer についてくるエンコーダです。mplayer をビルド (コンパイル) すると mencoder もついてきます。本来 CUI で動くものですが、GUI 版も配布されているので心配せずに。

QuickTime で再生したい場合には、avclencoder というプラグインを組み込んでやれば良いのですが、B フレームを使っているとまた厄介な問題が出てきます。ageha さんも苦勞しているようです。暫定的な対応策などが ageha さんのホームページに記載されていますので、それを参照してください。

後書き

どうだったでしょうか？ GUI で変換するよりもより高度でマニアックなオプションが使えたことかと思います。これを自分の好みで組み合わせることで、画質の限界に挑めるでしょう。本当は mencoder や x264 のビルド (コンパイル) 方法も解説したかったのですが、ちょっと部誌を書き始めるのが遅すぎましたね。というか、今はかなり切羽詰まった状況で書いています。締め切り 3 日前とかいうレベルを超えていますね。あとで怒られるかも…。まあそういうわけなので、多少表現がおかしかったとしても気にしないでください。これで mencoder や x264 に興味を持ってくれた方は ageha さんのホームページを参照すると良いです。というか、これ自体がホームページの内容を読みやすくまとめたような形になっていますので。それではまた会う日まで。

²⁾<http://www.squared5.com/>

⁴⁾<http://www.mplayerhq.hu/design7/news.html>

³⁾<http://www.videolan.org/vlc/>